



# <SPOTTUNES

Mario, Pedro, Ivan, Raul

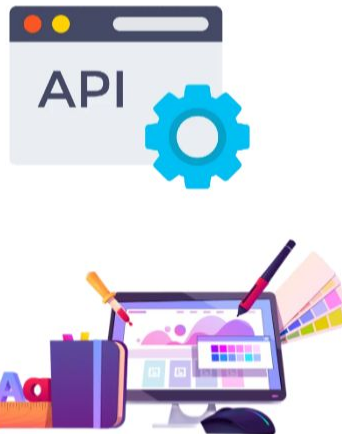
COMENÇAR>



## EVOLUCIÓ DE LES FUNCIONALITATS

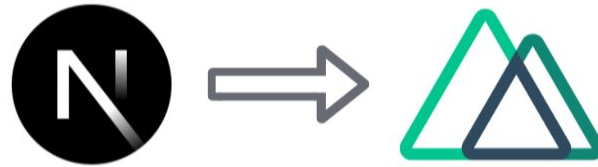
L'evolució de les funcionalitats al llarg dels Sprints ha estat notable. S'han completat diverses tasques importants en el desenvolupament del projecte, com la implementació de noves APIs, la correcció d'errors, el disseny i desenvolupament tant del front-end com del back-end, i la millora contínua del rendiment i la usabilitat de l'aplicació.

Cada Sprint ha suposat avanços significatius, des de la planificació i configuració inicial fins a la implementació i desplegament de funcionalitats clau, com la gestió de seguidors, la publicació de posts amb imatges, i la integració de serveis com Spotify i Google.

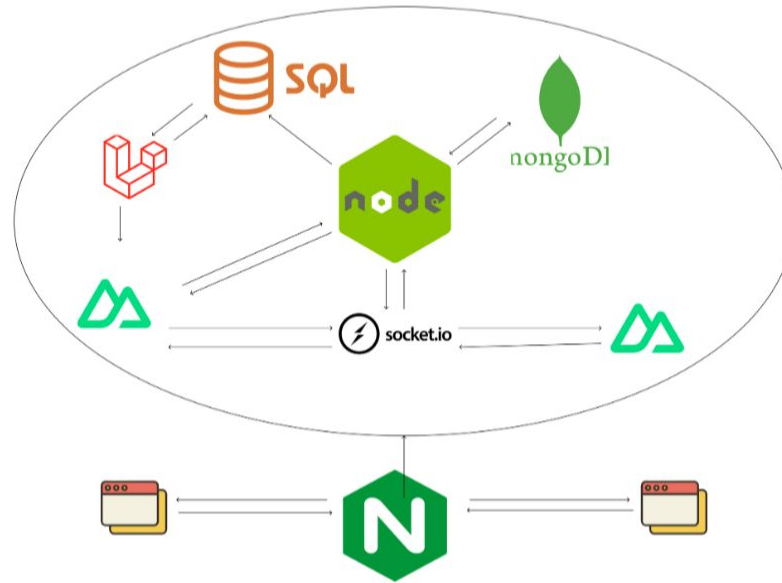


## PROBLEMES I SOLUCIONS

- Next
- Chats
- Proxy invers



## ASPECTES TÈCNICS



## REQUISITS TÈCNICS

### Màquina amb

- **8 GB** de RAM
- **4** cores
- **60 GB** d'emmagatzematge

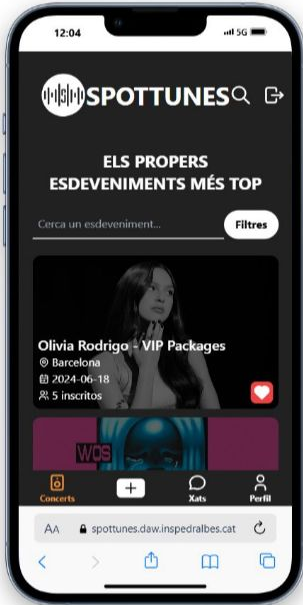


## FUNCIONALITATS PRINCIPALS

1. Concerts
2. Crear Posts
3. Xats
4. Perfil



## CONCERTS COMPONENT CARDEVENT



```
props: {
  event: {
    type: Object,
    required: true
  }
},
```

### Pages index.vue (all events)

```
eventosFiltrados: computed(() => this.store.events),
```

```
<div v-for="evento in eventosFiltrados" :key="evento.id">
  <CardEvent :event="evento" />
</div>
```

### Component CardEvent.vue

```
<div class="relative w-full h-56 rounded-xl overflow-hidden">
  <NuxtLink :to="store.getLoggedIn() ? '/events/${event.id}' : '/join'">
    
  <div class="absolute bottom-2 left-2">
    <h2 class="text-white text-xl font-semibold text-left">{{ event.event }}</h2>
    <p class="text-white flex items-center gap-1">
      <IconsMapPin class="size-4" />{{ event.city }}
    </p>
    <p class="text-white flex items-center gap-1">
      <IconsCalendar class="size-4" />{{ event.date }}
    </p>
    <p class="text-white flex items-center gap-1">
      <IconsUsers class="size-4" />{{ followers }} inscritos
    </p>
  </div>
</NuxtLink>

<button v-if="event.like" class="absolute bottom-2 right-2 p-1 rounded-lg bg-red-500 hover:bg-red-700"
  @click="togglelike">
  <IconsHeartFill size="20" />
</button>

<button v-else class="absolute bottom-2 right-2 p-1 rounded-lg bg-green-500 hover:bg-green-700"
  @click="togglelike">
  <IconsHeart size="20" />
</button>
```

## CONCERTS BUSCADOR USUARIOS



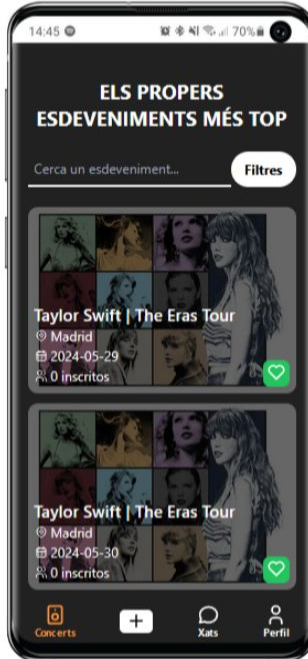
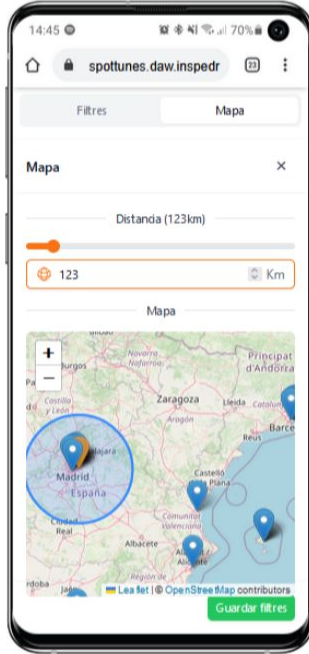
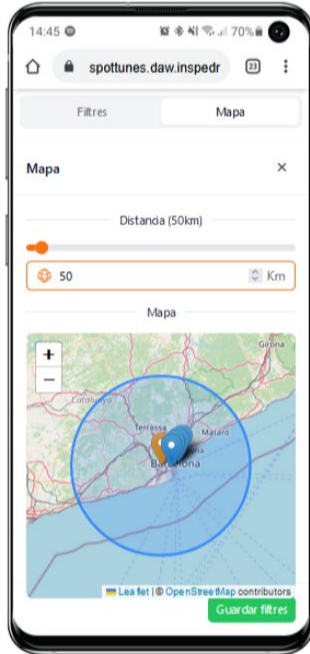
```
<USlideover v-model="sliderUsers" class="scroll-auto">
  <UCard class="flex flex-col flex-1"
    :ui="{ body: { base: 'flex-1' }, ring: '', divide: 'divide-y divide-gray-100 dark:divide-gray-800' }">
    <template #header>
      <div class="flex items-center justify-between">
        <h3 class="text-base font-semibold leading-6 text-gray-900 dark:text-white">
          Usuarios
        </h3>
        <UButton color="gray" variant="ghost" icon="i-heroicons-x-mark-20-solid" class="-my-1"
          @click="sliderUsers = false" />
      </div>
    </template>

    <div class="flex justify-center mt-9">
      <input v-model="param" @input="searchUsers" type="text" id="simple-search"
        class="bg-gray-50 border border-gray-300 text-black text-sm rounded-lg focus:ring-blue-500 focus:bor
        placeholder="Cerca usuarios..." required />
    </div>

    <!-- Lista de usuarios que devuelve el buscador -->
    <main class="flex flex-col justify-center items-center mt-8 space-y-4">
      <div v-if="!empty" v-for="user in filteredUsers" :key="user.id" class="w-[90%]">
        <CardUser class="text-black" :user="user" />
      </div>
      <div v-if="this.empty" class="flex justify-center items-center gap-1">
        <h2 class="font-bold m-5">{{ message }}</h2>
      </div>
    </main>
  </UCard>
</USlideover>
```



## CONCERTS FILTRES

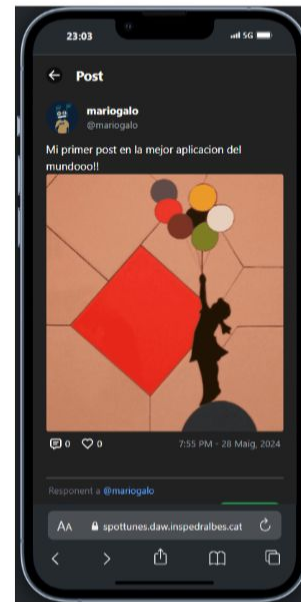
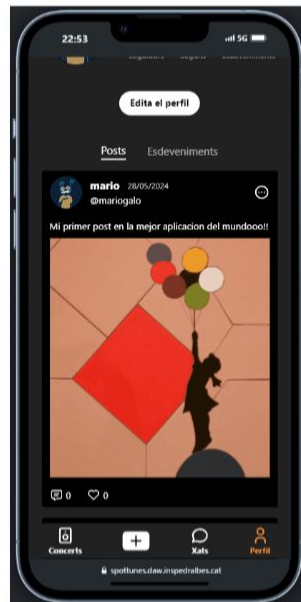


```
// obtenir les dades validades
latitude = $request->input('latitude');
longitude = $request->input('longitude');
distance = $request->input('distance');

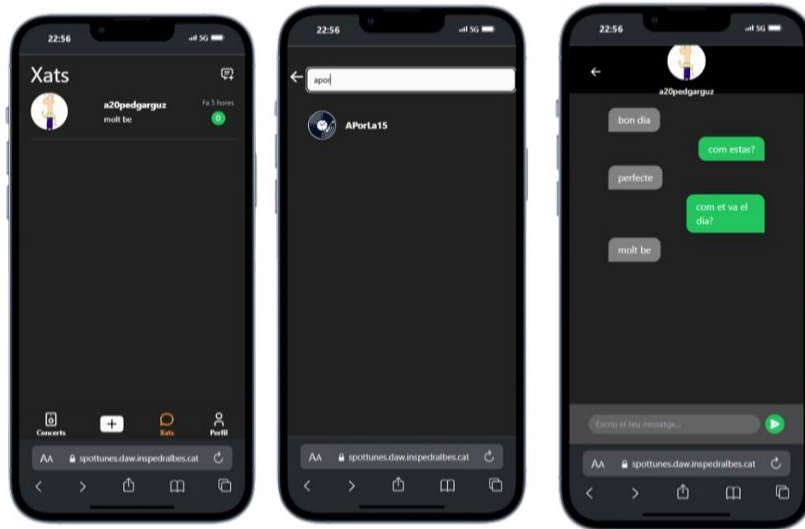
// realitzar la consulta
try {
    $events = Event::selectRaw('*', ( 6371 * acos( cos( radians(?) ) * cos( radians( latitude ) ) * cos(
        radians( longitude ) - radians(?) ) + sin( radians(?) ) * sin( radians( latitude ) ) ) ) AS distance',
        [latitude, longitude, $latitude])
        ->having('distance', '<= ', $distance)
        ->whereNotNull('artist')
        ->orderBy('date')
        ->orderBy('time')
        ->get();
} catch (\Exception $e) {
    return response()->json(['message' => 'Error fetching events', 'error' => $e->getMessage(), 500]);
}

// Verificar si hi ha esdeveniments
if ($events->isEmpty()) {
    return response()->json(['message' => 'No events found for the specified criteria'], 404);
}
```

## POSTS

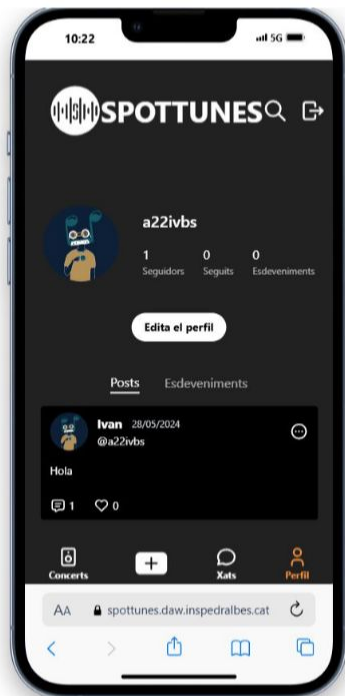


## XATS



```
1 socket.on("message", (message, contact) => {  
2   if (io.sockets.adapter.rooms.get(message.chat_id)?.size == 2) {  
3     message.status = "leido";  
4   } else if (userSockets.has(contact)) {  
5     message.status = "recibido";  
6   } else {  
7     message.status = "enviado";  
8   }  
9   manager.insertMessage(message).then((response) => {  
10    socket.join(response.chat_id);  
11  
12    io.to(response.chat_id).emit("message", response);  
13    if (io.sockets.adapter.rooms.get(response.chat_id)?.size != 2) {  
14      if (userSockets.has(contact)) {  
15        io.to(userSockets.get(contact)).emit("notification", response);  
16      } else {  
17        console.log("El usuario no está conectado");  
18      }  
19    }  
20  })  
21  .catch((error) => {  
22    console.error(error);  
23  });  
24  });  
25  });
```

## PERFIL



 No es poden escriure espais 





**Gràcies per la  
vostra atenció**

